



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/037,041	10/19/2001	Ravi Rajwar	960296.98440	5547
27114	7590	11/08/2005		
QUARLES & BRADY LLP 411 E. WISCONSIN AVENUE, SUITE 2040 MILWAUKEE, WI 53202-4497			EXAMINER TSAI, SHENG JEN	
			ART UNIT	PAPER NUMBER
			2186	

DATE MAILED: 11/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/037,041

Applicant(s)

RAJWAR ET AL.

Examiner

Sheng-Jen Tsai

Art Unit

2186

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 06 October 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-47 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-8, 11-20, 22-30, 33-41 and 43-47 is/are rejected.
- 7) ☒ Claim(s) 9-10, 21, 31-32 and 42 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- ☐ Notice of Informal Patent Application (PTO-152)
- ☐ Other: _____.

DETAILED ACTION

1. This Office Action is taken in response to Applicant's Amendments and Remarks filed on October 6, 2005 regarding application 10/037,041 filed on October 19, 2001.
2. Claims 1-47 are pending in the application under consideration.
3. ***Response to Amendment and Remarks***

Applicants' amendments and remarks have been fully and carefully considered with the results set forth below.

Response to Amendments and Remarks on Claims 1-2, 4, 7, 17-18, 22-23, 25, 28 and 38-39

Applicants explain in great detail the perceived differences between their invention and the disclosure by one of the references cited by the examiner (Mckeen et al., US 5,421,022). The examiner appreciates Applicants' effort on providing further information to distinguish the two inventions.

However, the wording and language of claims 1 and 22 do not distinctly recite the differences that Applicants describe in their remarks. It should be noted that if Applicant intends to distinguish their invention from that of Mckeen et al., these differences have to be reflected in the claim language.

Applicants also amended the claims by substituting the word "conflicts" with the word "interruption" in an effort to more clearly describe claimed invention.

However, this changing of wording does not alter, as far as claim analysis is concerned, the correspondence between the elements of the claims and the teaching by Mckeen et al., because within the context of Mckeen et al.'s disclosure a "conflict"

would always lead to an "interruption" to the successful completion of speculative executions.

Therefore, the examiner's position regarding the status of claims 1-2, 4, 7, 17-18, 22-23, 25, 28 and 38-39 remains the same as stated in the previous Office Action.

Response to Remarks on Claims 43-48

Applicant amended claim 43 with new, additional element of "wherein upon complete execution of the critical section without acquiring permission to write to the lock variable, if another thread has acquired the lock, squashing the execution of the critical section and then re-executing the critical region."

This new element is taught by one aspect of the invention disclosed by Chaudhry et al. (US 6,684,398) as indicated in column 2, lines 55-67, "upon receiving a request to perform a write operation by the head thread [i.e., execution of the critical section is completed and it is time to write the results to update the memory], the system performs the write operation to a primary version of the memory element. The system also checks status information associated with the memory element to determine if the memory element has been read by the speculative thread [i.e., if another thread has acquired the lock for accessing]. If the memory element has been read by the speculative thread, the system causes the speculative thread to roll back [i.e., squashing the execution of the critical section by the speculative thread] so that the speculative thread can read a result of the write operation [i.e., re-executing]. If the memory element has not been read by the speculative thread, the system performs the

write operation to a space-time dimensioned version of the memory element if the space-time dimensioned version exists.”

Therefore, the examiner’s position regarding the status of claims 43-48 remains the same as stated in the previous Office Action.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-2, 4, 7, 17-18, 22-23, 25, 28, and 38-39 are rejected under 35 U.S.C. 102(b) as being anticipated by McKeen et al. (US 5,421,022).

As to claim 1, McKeen et al. disclose **a method of coordinating access to common memory** [Apparatus and Method for Speculatively Executing Instructions in a Computer System (title)] **by multiple program threads** [figure 7 shows multiple program threads (71~74); column 6, lines 62-68; column 7, lines 1-3] **comprising the steps of:**

in each given program thread,

(a) detecting the beginning of a critical section of the given program thread in which interrupts to access of the common memory could occur resulting from execution of other program threads [a detection is made to decide if a set of instructions is in a real state or in a speculative state; the speculative state is the state where the dependency have not been resolved (column 5, lines 9-25); speculative

execution comprises two types of executions: data speculative and control speculative (column 5, lines 26-35)]

(b) speculatively executing the critical section [sets on instructions having unresolved dependencies are executed in a speculative state of the computer system under the assumption that an exception condition will not occur (abstract)]; and

(c) committing the speculative execution of the critical section if there has been no interruption to access of the common memory and squashing the speculative execution of the critical section if there has been an interruption [sets of instructions having unresolved dependencies are executed in a speculative state of the computer system under the assumption that an exception condition will not occur.

However, if an exception condition does occur while executing a set of instructions in the speculative state, that exception condition is detected and the set of instructions is re-executed in a real state of the computer system to resolve the exception condition (abstract)].

As to claim 2, McKeen et al. disclose that **the interruption is:**

(a) another thread writing data read by the given program thread in the critical section, or

(b) another thread reading or writing data written by the given program thread

[Examples are provided in figure 4 illustrating the interruptions between threads in terms of LOAD (read) and STORE (write) instructions; column 5, lines 45-66].

As to claim 4, McKeen et al. disclose that **the speculative execution is committed at the end of the critical section** [if an exception condition does occur

while executing a set of instructions in the speculative state, that exception condition is detected and the set of instructions is re-executed in a real state of the computer system to resolve the exception condition (abstract); a point is reached in the execution flow where it is known that the exception condition is resolved (column 4, lines 60-68; column 5, lines 1-2)].

As to claim 7, McKeen et al. disclose that **the speculative execution is committed at a resource boundary limiting further speculation** [general purpose data registers are used to store and perform data manipulation associated with speculative execution, such as exception flag and memory location (column 3, lines 10-15). Further, a register already allocated to a speculative thread will not be recycled until a point is reached in the execution flow where it is known that the exception condition is resolved (column 4, lines 60-67; column 5, lines 1-2). As such, the scope of speculative execution is limited by the number of registers available to the system, therefore bounded by the resource (i.e., the registers).

As to claim 17, McKeen et al. disclose that **the method of claim 1 including the further step of:**

(d) after squashing the speculative execution of the critical section if there has been a interruption, re-executing the critical section speculatively [if an exception condition does occur while executing a set of instructions in the speculative state, that exception condition is detected and the set of instructions is re-executed in a real state of the computer system to resolve the exception condition (abstract)].

As to claim 18, McKeen et al. disclose that **the speculative re-execution of the critical section is repeated up to a predetermined number of times until there is not a interruption** [a point is reached in the execution flow where it is known that the exception condition is resolved (column 4, lines 60-68; column 5, lines 1-2)].

As to claim 22, refer to "as to claim 1." Further, figures 10 and 11 show the circuit that facilitates the method and operations disclosed by McKeen et al.

As to claim 23, refer to "As to claim 2."

As to claim 25, refer to "As to claim 4."

As to claim 28, refer to "As to claim 7."

As to claim 38, refer to "As to claim 17."

As to claim 39, refer to "As to claim 18."

Claim Rejections - 35 USC § 102

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 43-47 are rejected under 35 U.S.C. 102(e) as being anticipated by Chaudhry et al. (US 6,684,398).

As to claim 43, Chaudhry et al. disclose **a method of coordinating the execution of multiple program threads executing critical sections of a program** [Monitor Entry and Exit for a Speculative Thread During Space and Time Dimensional Execution (title)], **the critical sections accessing common memory and being preceded by lock acquisition sections where writing to a lock variable mediates**

exclusive access to the common memory [a method of monitoring the entry into a critical section by a speculative thread, in which the entry is associated with the increment of a variable containing the number of virtual locks held by the speculative thread (abstract); and that each entry also contains a reference to a corresponding non-virtual lock that is held by a non-speculative thread; each entry 1212 in list 1210 contains a reference to a corresponding non-virtual lock 1214 that is held by a non-speculative head thread, such as other head thread 1204. Note that this non-virtual lock 1214 can be used to restrict access to an object 1216 and/or a critical section of code (column 9, lines 56-67)],

the method comprising the steps of:

for each given thread:

(a) upon reaching a lock acquisition section, reading the lock variable to see if the lock has been acquired by another thread [each entry also contains a reference to a corresponding non-virtual lock that is held by a non-speculative thread; each entry 1212 in list 1210 contains a reference to a corresponding non-virtual lock 1214 that is held by a non-speculative head thread, such as other head thread 1204. Note that this non-virtual lock 1214 can be used to restrict access to an object 1216 and/or a critical section of code (column 9, lines 56-67); and

(b) When at step (a) the lock has not been acquired by another thread, allowing the given thread to complete execution of the critical section without acquiring permission to write to the lock variable [Note that this non-virtual lock 1214 can be used to restrict access to an object 1216 and/or a critical section of code (column 9,

lines 56-67). In other words, if the lock has been acquired by other threads, the attempt to access the critical section by this thread is not allowed (restricted). If the lock has not been acquired by other threads, this thread is allowed to access the corresponding critical section. However, this thread can perform the speculative execution without acquiring the non-virtual lock by using the corresponding virtual lock (abstract)]; **and c) wherein upon complete execution of the critical section without acquiring permission to write to the lock variable** [upon receiving a request to perform a write operation by the head thread (i.e., execution of the critical section is completed and it is time to write the results to update the memory), the system performs the write operation to a primary version of the memory element (column 2, lines 55-67)], **if another thread has acquired the lock, squashing the execution of the critical section and then re-executing the critical region** [If the memory element has been read by the speculative thread, the system causes the speculative thread to roll back (i.e., squashing the execution of the critical section by the speculative thread) so that the speculative thread can read a result of the write operation (i.e., re-executing). If the memory element has not been read by the speculative thread, the system performs the write operation to a space-time dimensioned version of the memory element if the space-time dimensioned version exists].

As to claim 44, Chaudhry et al. teach that **the lock variable is held within a cache and wherein acquiring permission to write to the lock variable includes the step of obtaining ownership of at least a portion of the cache holding the lock variable** [acquiring and releasing a lock may cause a cache miss (i.e., the lock is

stored within a cache) and may require load buffer and/or store buffer to be flushed. In other words, acquiring and releasing a lock requires a store (write) operation to the cache memory address where the lock variable resides in order to update the status of the lock, including obtaining the ownership of the lock].

As to claim 45, refer to "As to claim 43."

As to claim 46, refer to "As to claim 44." Note that acquiring the access to a cache memory location where the lock variable is stored may result in a cache miss and a chain of subsequent operations (such as replacement of victim, write-through or write-back, cache coherency, etc.), which are all cache protocol message.

As to claim 47, Chaudhry et al. teach that **during step (b) the execution is not committed and including the further step of:**

(d) when during step (b) the resources required for execution without commitment are exhausted, acquiring the lock, committing the execution and continuing with execution until completion of the critical section [this is accomplished by the use of both the non-virtual locks and the virtual locks (figure 15)].

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 5-6, 8, 11, 13-16, 19, 26-27, 29-30, 34-37 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over McKeen et al. (US 5,421,022), and in view of Chaudhry et al. (US 6,684,398).

As to claim 5, McKeen et al. do not teach **the end of the critical section is detected by a pattern of instructions typically associated with a lock release.**

However, it is a well-known and common practice in the art to associate the entry and exit of a critical section with acquiring and releasing a lock, respectively, in order to resolve conflicts for accessing shared resources.

Further, Chaudhry et al. teach in their invention "Monitor Entry and Exit for a Speculative Thread During Space and Time Dimensional Execution" a method of monitoring the entry into a critical section by a speculative thread, and that acquiring (signaling entry) and releasing (signaling exit) a lock is a mechanism commonly adopted to restrict access to critical sections (column 1, lines 56-67).

Therefore, it would have been obvious for ones of ordinary skills in the art at the time of Applicants' invention to recognize the well-known and common practice of associating the entry and exit of a critical section with acquiring and releasing a lock, respectively, as demonstrated by Chaudhry et al., and its lacking patentable significance.

As to claim 6, refer to "As to claim 5." Further, Chaudhry et al. teach that acquiring and releasing a lock may cause a cache miss and may require load buffer and/or store buffer to be flushed. In other words, acquiring and releasing a lock requires

a store (write) operation to the address where the lock variable resides in order to update the status of the lock.

As to claim 13, McKeen et al. do not teach **deducing the beginning of a critical section by detecting patterns of instructions typically associated with a lock acquisition.**

However, Chaudhry et al. teach in their invention "Monitor Entry and Exit for a Speculative Thread During Space and Time Dimensional Execution" a method of monitoring the entry into a critical section by a speculative thread, in which the entry is associated with the increment of a variable containing the number of virtual locks held by the speculative thread (abstract); and that each entry also contains a reference to a corresponding non-virtual lock that is held by a non-speculative thread (column 9, lines 56-67).

The use of a virtual lock and the reference to the corresponding non-virtual lock held by other threads provides an effective way of coordinating operations among all threads to prevent interruptions from happening.

Therefore, it would have been obvious for ones of ordinary skills in the art at the time of Applicants' invention to recognize the benefits of using virtual locks and non-virtual locks to facilitate the coordination among threads, as demonstrated by Chaudhry et al., and to incorporate it into the existing scheme disclosed by McKeen et al. to further enhance the performance of the system.

As to claim 8, refer to "As to claim 13." Further, Chaudhry et al. teach that the corresponding non-virtual lock can be used to restrict access to a critical section if an

interruption exits, and allows the execution to continue if there is no interruption (column 9, lines 56-67).

As to claim 11, refer to "As to claim 8."

As to claim 14, refer to "As to claim 13." Further, an atomic read/modify/write sequence is a well-known concept and practice commonly adopted in the art of computer system (refer to Microsoft Computer Dictionary, 5th edition, Microsoft Press, 2002, page 40), hence lacks patentable significance and is not patentable.

As to claim 15, refer to "As to claim 13." Further, Chaudhry et al. teach that the virtual lock does not prevent the speculative thread or other threads from entering the critical section, hence essentially eliding the lock acquisition.

As to claim 16, refer to "As to claim 13." Further, Chaudhry et al. teach that the speculative execution of a critical section ends with the receiving of a request to perform a joining operation, and the speculative thread waits until the variable containing the number of virtual locks held by the speculative thread equals zero.

As to claim 19, refer to "As to claim 13." Further, Chaudhry et al. teach that the corresponding non-virtual lock can be used to restrict access to a critical section if a interruption exits, and allows the execution to continue if there is no interruption (column 9, lines 56-67).

As to claim 26, refer to "As to claim 5."

As to claim 27, refer to "As to claim 6."

As to claim 29, refer to "As to claim 5" and figure 15 of Chaudhry et al.

As to claim 30, refer to "As to claim 11."

As to claim 34, refer to "As to claim 13."

As to claim 35, refer to "As to claim 14."

As to claim 36, refer to "As to claim 15."

As to claim 37, refer to "As to claim 16."

As to claim 40, refer to "As to claim 19."

9. Claims 3, 20, 24 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over McKeen et al. (US 5,421,022), and in view of Shibayama et al. (US Patent Application Publication 2003/0014602).

As to claim 3, McKeen et al. do not teach that **the interruption is detected by an invalidation of a cache block holding data of the critical section.**

However, Shibayama et al. teach in their invention "cache Memory Control method and Multi-Processor System" a scheme of performing multiple threads speculative executions with the use of cache memory, in which an effective flag is associated with every cache line to indicate whether the cache line is effective or invalid (i.e., presence of interruptions) as a result of speculative execution (paragraph 0045).

The use of such flag is crucial in maintaining the coherency of the memory system I support of speculative execution.

Therefore, it would have been obvious for ones of ordinary skills in the art at the time of Applicants' invention to recognize the benefits of indicating the invalidation of a cache line to signal the presence of interruptions of speculative execution as well as maintaining the memory coherency, as demonstrated by Shibayama et al., and to

incorporate it into the existing scheme disclosed by McKeen et al. to further ensure the integrity of the memory system.

As to claim 20, Shibayama et al. teach that **the speculation executes the critical section using a cache memory to record the speculative execution without visibility to other processing units** [figures 9 and 10 show the use of cache memory in support of the speculative executions].

As to claim 24, refer to "As to claim 3."

As to claim 41, refer to "As to claim 20."

10. Claims 12, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over McKeen et al. (US 5,421,022), and in view of Shibayama et al. (US Patent Application Publication 2002/0178349).

As to claim 12, McKeen et al. do not teach that **step (a) includes reading a prediction table holding historical data indicating past successes in speculatively executing the critical section and wherein step (b) is performed only when the prediction table indicates a likelihood of successful speculative execution of the critical section of above a predetermined threshold.**

However, Shibayama et al. teach in their invention "Processor, Multiprocessor System and Method for Data Dependency Speculative Execution" a scheme of performing multiple threads speculative executions using predictions based on a history table which stores history information concerning success/failure results of the speculative execution of memory operation instructions of the past. If the prediction is

“success”, speculative execution is performed; if the prediction is “failure”, the speculative execution is canceled (abstract).

The use of such a prediction scheme based on the past success/failure history leads to better efficiency and increases the throughput.

Therefore, it would have been obvious for ones of ordinary skills in the art at the time of Applicants’ invention to recognize the benefits of such a prediction scheme, as demonstrated by Shibayama et al., and to incorporate it into the existing scheme disclosed by McKeen et al. to further improve the throughput of the system.

As to claim 33, refer to “As to claim 12.”

Allowable Subject Matter

11. Claims 9-10, 21, 31-32 and 42 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

12. *Related Prior Art*

The following list of prior art is considered to be pertinent to applicant’s invention, but not relied upon for claim analysis conducted above.

- Ohsawa et al., (US Patent Application Publication 2003/0014473), “Multi-Thread Executing Method and Parallel Processing System.”

Conclusion

13. Claims 1-8, 11-20, 22-30, 33-41 and 43-47 are rejected as explained above.

Claims 9-10, 21, 31-32 and 42 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

14. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.


15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sheng-Jen Tsai
Examiner
Art Unit 2186

October 31, 2005


PIERRE BATAILLE
PRIMARY EXAMINER
11/03/05